SISTEMAS DE INFORMACIÓN II

(IF202)

UN APUNTE PRELIMINAR

Separata 3: Fases

Ingº Manuel A. Peñaloza Figueroa

Departamento Académico de Informática

Universidad Nacional de San Antonio Abad del Cusco

Cusco, Perú

En Informática, nada es absoluto, todo es relativo.

Leunam.

3 FASES

Fase es esencialmente un lapso de tiempo entre 2 hitos/milestones mayores o importantes, durante el cual:

- → un conjunto bien-definido de objetivos es satisfecho
- → artefactos son completados
- → (y) decisiones son tomadas para moverse o no dentro de la siguiente fase.

- † Al final de cada fase una valoración es llevada a cabo para determinar si los objetivos de la fase han sido satisfechos.
- † Una valoración satisfactoria permite al proyecto moverse a la siguiente fase.

Fases e Hitos

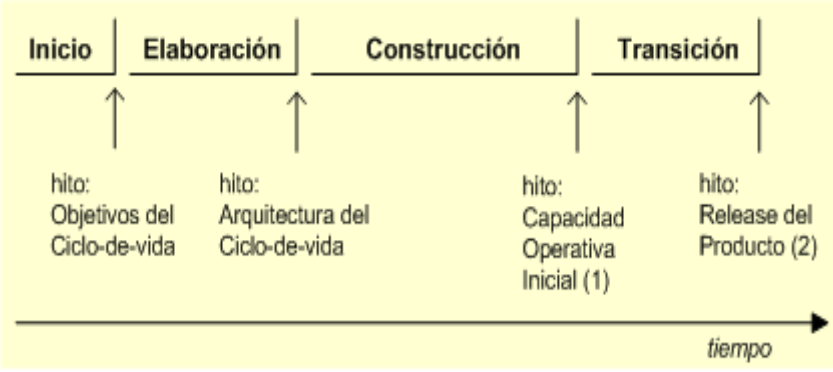


Figura: PU: Fases e hitos de un proyecto.

Planificando las Fases

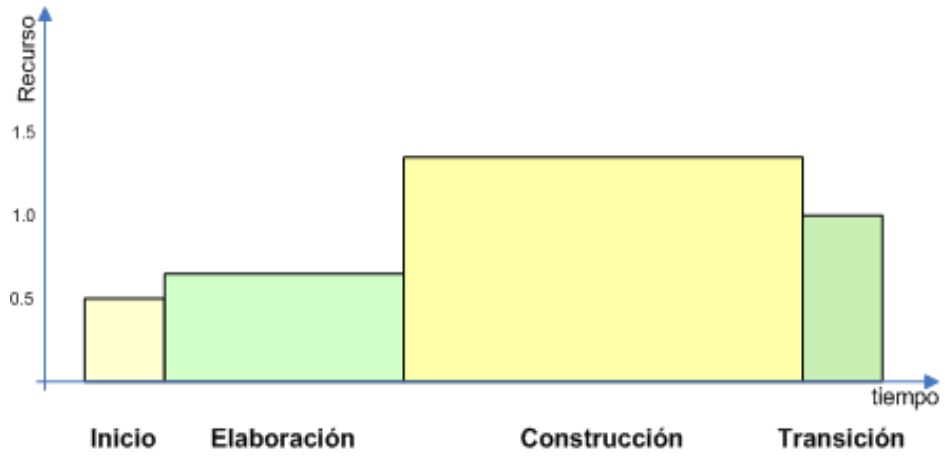
Todas las fases no son idénticas en términos de cronograma y esfuerzo.

- Esto varía considerablemente dependiendo del proyecto.

Un ciclo de desarrollo típico inicial para un proyecto de mediano tamaño debe de anticipar la siguiente distribución entre esfuerzo y cronograma:

	Inicio	**Elaboración**	**Construcción**	**Transición**
Esfuerzo	~5 %	20 %	65 %	10 %
Cronograma	10 %	30 %	50 %	10 %

El cual puede ser descrito gráficamente como:



Artefactos de Requerimientos planificados[1]

Disciplina	Artefacto	I	E	C	T	Responsable
R.	Modelo de Caso de Uso	I	M			Analista de Sistemas
	Prototipo IU		I			Diseñador IU
	Especificaciones Suplementarias	I	M			Analista de Sistemas
	Visión	I	M			Analista de Sistemas

	I	E	C	T
Modelo del Dominio	I	F		
Modelo del Negocio	I	F		
Requerimientos	I	F		
Análisis	I	F		
Arquitectura	I	A	F	
Lista de Riesgos	I	A	A	

[1] Fuente: Adaptado de: "Software Development for Small Teams: A RUP-Centric Approach"; Gary Pollice, Liz Augustine, Chris Lowe & Jas Madhur; Chapter 4, Getting Started: The Project Members Become a Team, pp. 48.

Casos de Uso	I			
Plan para la fase de E.	C			
Caso del Negocio	I	F	a	
Plan de Administración del Proyecto		C	A	
Manuales			I	F
Release			B	

3.1 Fase: Inicio

3.1.1 Meta

La meta primordial de la fase de Inicio es:

- Lograr la coincidencia entre todos los interesados sobre los objetivos del ciclo-de-vida para el proyecto.

> "La fase de Inicio es de significación primariamente para nuevos esfuerzos de desarrollo, en los cuales hay **significantes riesgos de negocio y requerimientos** los cuales tienen que ser abordados antes de que el proyecto pueda proseguir.
>
> - Para proyectos enfocados sobre mejoramientos a un sistema existente, la fase de Inicio es más breve, pero está todavía enfocado en asegurar que el proyecto tanto vale hacer como que es posible hacerlo.

3.1.2 Objetivos:

Objetivos primarios:

- **Establecer el alcance de software del proyecto**: (y las condiciones de limite)
  Incluyendo:
  - → una visión operativa
  - → criterios de aceptación
  - → (y) que está intentado estar en el producto y lo que no esta.
- **Discriminar los casos de uso críticos del sistema**: (y los escenarios primarios de operación que conducirán los mayores trade-offs de diseño).
- **Mostrar, al menos una arquitectura candidata**: contra alguno de los escenarios primarios.
- **Estimar el costo global y el cronograma para el proyecto entero**: (y estimados más detallados en la fase de Elaboración que inmediatamente sigue).
- **Estimar riesgos potenciales**: (las fuentes de impredicibilidad).
- **Preparar el entorno de soporte para el proyecto**.

3.1.3 Actividades esenciales:

- **Formular el alcance del proyecto**: Esto involucra capturar el contexto y los más importantes requerimientos y restricciones hasta el punto de que se puede derivar los criterios de aceptación para el producto final.

- **Planificar y preparar un caso de negocio**: Evaluando alternativas para:
  - → administración del riesgo
  - → la provisión ó dotacion de personal
  - → plan de proyecto
  - → (y) los intercambios de una cosa a cambio de otra/tradeoffs entre costo/cronograma/rentabilidad

Caso de Negocio:

El Caso de Negocio provee la información necesaria desde un punto de vista del negocio para determinar si o no en el proyecto vale la pena invertir.

Para un producto de software comercial, el Caso de Negocio debe de incluir un conjunto de suposiciones acerca del proyecto y el orden de magnitud del Retorno de Inversión (ROI) si esas suposicones son ciertas.

- Por ejmplo, el ROI será de una magnitud de 5 si completado en un año, 2 si completado en 2 años, y un número negativo después de eso.
- Estas supociones son chequeadas de nuevo al final de la fase de Elaboración, cuando el alcance y el plan están definidos con más exactitud.

- **Sintetizar una arquitectura candidata**: evaluando trade-offs:
  - → en el diseño
  - → (y) en el hacer/comprar/ reusar

  de modo que:
  - → el costo
  - → el cronograma
  - → (y) los recursos

  puedan ser estimados.

- La intención/propósito aquí es demostrar viabilidad/factibilidad a través de alguna clase de prueba de conceptos.
- Esto puede tomar la forma:
  - → de un modelo el cual simula que es requerido
  - → (o) de un prototipo inicial el cual explora que se considera que son las áreas de alto riesgo.
- El esfuerzo de prototipeo durante el Inicio debe de estar limitado a ganar confianza que una solución es posible – la solución es realizada durante la Elaboración y la Construcción.

- **Preparar el entorno para el proyecto**:
  - → valorando el proyecto y la organización
  - → seleccionando las herramientas
  - → decidiendo cuales partes del proceso mejorar

3.1.4 Hito: Objetivos del Ciclo-de-vida

El hito Objetivos del Ciclo-de-vida evalua la viabilidad básica del proyecto.

- En este punto, se examina los objetivos del ciclo-de-vida del proyecto, y se decide si proseguir con el proyecto o cancelarlo.

Criterio de Evaluación

- Concurrencia de los interesados sobre la definición del alcance y los estimados de costo/cronograma.
- Acuerdo que el conjunto correcto de requerimientos han sido capturados y que hay un entendimiento compartidos de estos requerimientos.
- Acuerdo que:
  - → los estimados de costo/cronograma
  - → prioridades
  - → riesgos
  - → (y) proceso de desarrollo

  son apropiados.
- Todos los riesgos han sido identificados y una estrategia de mitigación existe para cada una.

El proyecto puede ser abortado o considerablemente re-pensado si falla en alcanzar este hito.

3.1.5 Artefactos (I)

Artefactos esenciales **(en orden de importancia)**	**Estado en el hito**
Visión	→ los requerimientos "core" → las caractereísticas claves → (y) las restricciones principales del proyecto están documentados.
RUP: Caso de Negocio	Definido y aprobado.
Lista de Riesgos	Riesgos iniciales del proyecto identificados.

RUP: Plan de Desarrollo de Software	Fases iniciales: → su duración → (y) objetivos identificados. • Estimados de recursos (específicamente el tiempo, staff, y costos del entorno de desarrollo en particular) en el Plan de Desarrollo de Software tienen que ser consistentes con el Caso de Negocio. El estimado de recursos puede abarcar ó el proyecto entero a través de entregables, ó solo un estimado de recursos necesitados para ir a través de la fase de Elaboración. • Estimados de los recursos requeridos para el proyecto entero deben de ser vistos como muy vagos/aproximados, un "calculo al ojo" en este punto. • Este estimado es actualizado en cada fase y cada iteración, y se vuelve más preciso con cada iteración. Dependiendo de las necesidades del proyecto, uno o más de los artefactos del "Plan" adjuntado pueden estar condicionalmente completados. • Un inicial Plan de Aceptación del Producto debe de ser revisado y con línea-de-base. • El Plan de Aceptación del Producto es refinado en iteraciones subsecuentes a medida que requerimientos adicionales son descubiertos. Además, los artefactos "Líneas-de-guía" adjuntados están típicamente en al menos una forma de "borrador".
Plan de Iteración	El plan de iteración para la primera iteración de Elaboración completado y revisado.
RUP: Proceso de Desarrollo	Adaptaciones y extensions al RUP, documentadas y revisadas. • Esto típicamente incluye líneas-de-guía y

	plantillas, así como un caso de desarrollo para documentar decisiones de personalización/adaptación específicas al proyecto.
RUP: Infraestructura de Desarrollo	Todas las herramientas para soportar el proyecto son seleccionadas. • Las herramientas necesarias para trabajar en Inicio están instaladas. En particular, el entorno de Administración de la Configuración debe de estar "set up".
Glosario	Términos importantes definidos; glosario revisado.
Modelo de Caso-de-Uso (Actores y Casos de Uso)	→ Actores y casos de uso importantes identificados → (y) flujo de eventos esbozados para solo los más críticos casos de uso.
UPEDU: Depositario del Proyecto, Petición/Request de Cambio	El entorno de Administración de la Configuración debe de estar "set up". (*)
Artefactos Opcionales	
RUP: Modelo de Dominio (Modelo de Análisis de Negocio a.k.a.)	Los conceptos claves siendo usados en el sistema, documentados y revisados. • Usado como una extension al Glosario en casos donde hay (inter)relaciones específicas entre conceptos que son esenciales capturar.
RUP: Prototipos	Uno o más prototipos de prueba de conceptos, para: → soportar la Visión y el Caso de Negocio → (y) abordar riesgos muy específcos.

(*) Parte de este artefacto se encuentra en el artefacto del RUP: Infraestructura de desarrollo.

Depositario del Proyecto:

El Depositario del Proyecto almacena todas las versiones de archivos y directorios del proyecto.

- También almacena todos los datos derivados y meta datos asociados con los archivos y directorios.

3.2 Fase: Elaboración

3.2.1 Meta:

La meta del la fase de Elaboración es:

→ desarrollar la línea-de-base de la arquitectura del sistema para proveer una base estable para el grueso del esfuerzo de diseño e implementación en la fase de Construcción.

"La arquitectura evoluciona más alla de:

→ una consideración de los más significantes requerimientos (aquellos que tienen un gran impacto sobre la arquitectura del sistema)

→ (y) de una valoración del riesgo.

- La estabilidad de la arquitectura es evaluada a través de uno o más prototipos arquitecturales.

3.2.2 Objetivos:

Los objetivos primarios de la fase de Elaboración incluyen:

- **Asegurar que la arquitectura, requerimientos y planes son lo suficientemente estables y los riesgos suficientemente mitigados** para poder predeciblemente determinar el costo y el cronograma para la conclusión/completion del desarrollo.
  - Para la mayor parte de los proyectos, pasar este hito también corresponde a la transición desde una operación ligera-y-rápida, de bajo riesgo a una operación de alto costo y, alto riesgo con sustancial inercia organizacional.
- **Abordar todos los riesgos arquitecturalmente significantes del proyecto**.
- **Establecer una arquitectura con línea-de-base** derivada a partir de abordar los escenarios arquitecturalmente significantes, los cuales típicamente exponen/sacan a la luz los riesgos técnicos más altos del proyecto.
- **Producir un prototipo evolutivo de componentes de calidad-de-producción**, así como posiblemente uno o más prototipos exploratorios, desechables/descartables para mitigar riesgos específicos tales como:
  - † trade-offs de diseño/requerimientos
  - † reuso de componentes
  - † viabilidad del producto ó demostraciones a:
    - → los inversionistas
    - → los clientes
    - → (y) usuarios finales.

- **Demostrar que la arquitectura con línea-de-base soportará lo requerimientos del sistema a un costo razonable y en un tiempo razonable**.
- **Establecer un entorno de soporte**.

  A fin de lograr estos objetivos primarios, es igualmente importante "set up" el entorno de soporte para el proyecto.
  - Esto incluye:
    - → crear un caso de desarrollo
    - → crear plantillas
    - → líneas-de-guía
    - → (y) "setting up" herramientas.

3.2.3 Actividades esenciales

- **Definir, validar y desarrollar la línea-de-base/baselining de la arquitectura**: tan rápidamente como práctica.
- **Refinar la Visión**: basado sobre nueva información obtenida durante la fase, estableciendo un entendimiento sólido de los más críticos casos de uso que conducen las decisiones arquitecturales y de planificación.
- **Crear y desarrollar la línea-de-base**: para los planes de iteración detallados para la fase de Construcción.
- **Refinar el caso de desarrollo y colocar en el lugar el entorno de desarrollo**: incluyendo el proceso, el soporte de herramientas y automatización requeridos para soportar al equipo de construcción.
- **Refinar la arquitectura y seleccionar componentes**: Componentes potenciales son evaluados y las decisiones de hacer/comprar/reusar suficientemente entendidas para determinar el costo y cronograma de la fase de Construcción con confianza/confidence.
  - Los componenentes arquitecturales seleccionados son integrados y valorados contra los escenarios primarios.
  - Lecciones aprendidas desde estas actividades pueden bien resultar en un rediseño de la arquitecura, tomando en consideración diseños alternativos o reconsideración de los requerimientos.

Caso de Desarrollo:

El Caso de Desarrollo describe el proceso de desarrollo que se ha elegido para seguir el proyecto.

El Caso de Desarrollo describe como el proceso es aplicado para un proyecto

específico ú organización.

Incluye detalles como:

- que actividades y artefactos opcionales serán usado, y cuales serán excluídos.
- el cronometraje relativo de actiidades para cada fase.
- que herramientas serán usudas.
- el nivel de formalidad a ser aplicado.

3.2.4 Hito: Arquitectura del Ciclo-de-Vida

El hito Arquitectura del Ciclo-de-Vida:

→ establece una línea-de-base administrada por la arquitectura del sistema

→ (y) habilita al equipo del proyecto a escalar durante la fase de Construcción.

- En este punto, se examina:
  - → los objetivos detallados del sistema y el alcance
  - → la selección de la arquitectura
  - → (y) la resolución de los mayores riesgos.

Criterio de Evaluación

- El producto Visión y los requerimientos son estables.
- La arquitectura es estable
- Los enfoques claves a ser usados en la prueba y evaluación están probados.
- La prueba y evaluación de prototipos ejecutables han demostrado que los mayores elementos de riesgo han sido abordados y han sido creíblemente resueltos.
- Los planes de iteración para la fase de Construcción son de suficiente detalle y fidelidad para permitir que el trabajo prosiga.
- Los planes de iteración para la fase de Construcción están soportados por estimados creibles.
- Todos los interesados están de acuerdo que la visión actual puede ser satisfecha si el plan en curso es ejecutado para desarrollar el sistema completo, en el contexto de la arquitectura actual.
- Gasto de recursos actual Vs gasto planificado son aceptables.

El proyecto puede ser abortado o considerablemente re-pensado si se falla en alcanzar este hito.

"El sistema "flaco" evoluciona para volverse el sistema completamente desarrollado, tal vez con algunos cambios menores a la estructura y al

comportamiento.

- Los cambios son menores por que:

  {Al final de la Elaboración o iteraciones arquitecturales, se tiene por definición una arquitectura estable};

  de otra manera,

  {La fase de Elaboración tiene que continuar hasta que se sepa que esta meta ha sido lograda}.

- Hay un modo sistemático de hacer esto:

  Al final de la fase de Elaboración:

  → se han desarrollado modelos del sistema que representan los casos de uso más importantes y sus realizaciones, desde la perspectiva de la arquitectura.

  → se han decidido:

  † con que estándares se cuenta

  † que software del sistema y que middleware utilizar

  † que sistema heredados reutilizar

3.2.5 Artefactos (E)

Artefactos Esenciales (en orden de importancia)	**Estado en el hito**
RUP: Prototipos	Uno o más prototipos arquitecturales ejecutables han sido creados para explorar: → funcionalidad crítica → (y) escenarios arquitecturalmente significantes.
Lista de Riesgos	Actualizado y revisado. • Nuevos riesgos son probablemente ser arquitecturales en naturaleza, en primer lugar relacionaldos al manejo de requerimientos no-funcionales.
RUP: Proceso de Desarrollo	El proceso de desarrollo, incluyendo cualesquier líneas-de-guía y plantillas/ específicos al proyecto, han sido refinados basados en la experiencia de los primeros proyectos, y está suficientemente definido para la fase de Construcción proseguir.
RUP: Infraestructura de Desarrollo	El entorno de desarrollo para construcción está en su lugar, incluyendo todas las herramientas y soporte de automatización para el proceso.

Documento de Arquitectura del Software	Creado y con línea-de-base, incluyendo descripciones detalladas para los casos de uso arquitecturalmente significantes (vista de caso-de-uso), identificación de mecanismos claves y elementos de diseño (vista lógica), más la definición de la vista del proceso y la vista de desplegamiento si el sistema está distribuído o tiene que tratar con temas de concurrencia.
Modelo de Diseño (y todos los artefactos constituyentes)	Definido y con línea-de-base. • Realizaciones de Caso-de-uso para escenarios arquitecturalmente significantes han sido definidos y comportamiento requerido ha sido asignado a los elementos de diseño apropiados. • Componentes han sido identificados y las desiciones de hacer/comprar/reusar suficientemente entendidas para determinar el costo y el cronograma de la fase de Construcción con confianza. • Los componentes arquitecturales seleccionados son integrados y valorados contra los escenarios primarios. • Lecciones aprendidas a partir de estas actividades pueden bien resultar en un rediseño de la arquitectura, tomando en consideración diseños alternativos o reconsideración de los requerimientos.
RUP: Modelo de Datos	Definido y con línea-de-base. • Los mayores elementos del modelo de datos (e.g. entidades importantes, (inter)relaciones, tablas) definidos y revisados.
Modelo de Implementación (y todos los artefactos constituyentes, incluyendo Componentes (ó Elementos de Implementación))	Estructura inicial creada y mayores components identificados y prototipeados.
Visión	Refinado, basado sobre nueva información obtenida durante la fase, estableciendo un entendimiento sólido de los más críticos casos de uso que conducen las decisiones arquitecturales y de planificación.
RUP: Plan de Desarrollo de Software	Actualizado y expandido para cubrir las fases de Construcción y de Transición.
Plan de Iteración	Plan de iteración para la fase de Construcción

	completado y revisado.
Modelo de Caso-de-Uso (Actores, Casos de Uso)	Un modelo de caso-de-uso (aproximadamente **80% completo**): → todos los casos de uso habiendo sido identificados en la encuesta/survey del modelo de caso-de-uso → todos los actores habiendo sido identificados → (y) la mayor parte de las descripciones de caso-de-uso (captura de requerimientos) han sido desarrollados.
Especificaciones Suplementarias	Requerimientos suplementarios capturando los requierimientos no-funcionales están documentados y revisados.
RUP: Suite de Pruebas ("smoke test")	Pruebas implementadas y ejecutadas para validar la estabilidad del build para cada uno de los releases ejecutables creados durante la fase de Elaboración. ???????
RUP: Arquitectura de Automatización de Pruebas/Test Automation Architecture	Una composición con base-de-linea de varios mecanismos y elementos de software claves que plasman las características fundamentales del sistema de software de automatización de pruebas.
Artefactos Opcionales	**Estado en el hito**
RUP: Caso de Negocio	Actualizado si investigaciones arquitecturales destapan cuestiones que cambien fundamentales suposiciones del proyecto.
RUP: Modelo de Análisis	Puede ser desarrollado como un artefacto formal; ni frecuentemente ni formalmente mantenido, evolucionando en una primera versión del Modelo de Diseño más bien.
RUP: Material de Soporte del Usuario-Final	Manuales de Usuario y otros materials de entrenamiento. • Preliminarmente borrador, basado en casos de uso. • Puede ser necesitado si el sistema tiene un fuerte aspecto de interfase del usuario.

3.3 Fase: Construcción

3.3.1 Meta

La meta de la fase de construcción es:

- → clarificar los requerimientos restantes
- → (y) completar el desarrollo del sistema basado sobre la arquitectura con línea-de-base.

> "La fase de Construcción es en algún sentido un proceso de manufacturación, donde énfasis es colocado sobre:
>
> → la administración de recursos
>
> → (y) el control de las operaciones
>
> para optimizar:
>
> → costos
>
> → cronogramas/schedules
>
> → (y) calidad.
>
> • En este sentido el esquema mental de la administración sufre/experimenta una transición desde el desarrollo de propiedad intelectual durante el Inicio y la Elaboración, al desarrollo de productos desplegables durante Construcción y Transición".

3.3.2 Objetivos:

Objetivos primarios:

- **Minimizar los costos de desarrollo** con optimizar recursos y evitar innecesario scrap y retrabajo.
- **Conseguir/lograr adecuada calidad** tanto rápidamente como práctica.
- **Conseguir/lograr versiones útiles** (alpha, beta, y otros releases de prueba) tanto rápidamente como práctica.
- **Completar el análisis, diseño, desarrollo y prueba** de toda la funcionalidad requerida.
- **Iterativamente e incrementalmente desarrollar un producto completo** que está listo para transición a su comunidad de usuarios.
  - Esto implica:
    - → describir los casos de uso restantes y otros requerimientos
    - → elaborar el diseño
    - → completar la implementación
    - → (y) probar el software

- **Decidir si:**
  - → **el software**
  - → **los sites**
  - → **(y) los usuarios**

  están todos listos para que la aplicación sea desplegada.

- **Conseguir algún grado de paralelismo en el trabajo de los equipos/teams de desarrollo**.
  - Aún en proyectos más pequeños, hay típicamente componentes que pueden ser desarrollados independientemente uno del otro, dejando un margen para paralelismo natural entre equipos/teams (resources permitting).
  - Este paralelismo puede acelerar las actividades de desarrollo significativamente; pero también incrementa:
    - → la complejidad de la gestión de recursos
    - → (y) la sincronización del flujo-de-trabajo.
  - Una arquitectura robusta es esencial si algún paralelismo significante está por ser conseguida.

3.3.3 Actividades Esenciales:

Las actividades esenciales de la fase de Construcción incluyen:

- **Gestión/Administración de recursos, control y optimización del proceso**.
- **Completar desarrollo de componentes y pruebas** contra los criterios de evaluación definidos.
- **Valoración de releases del producto** contra criterios de aceptación para la visión ???.

3.3.4 Hito: Capacidad Operativa Inicial

El hito Capacidad Operativa Inicial determina si el producto está listo para ser desplegado dentro un entorno de prueba-beta.

- El producto es listo para ser transferido al Equipo/Team de transición.
- Toda la funcionalidad ha sido desarrollada y todas las prueba alpha (si alguna) ha sido completada.
- Además al software, un manual de usuario ha sido desrarrollado, y hay una descripción del release actual.

Criterio de Evaluación:

El criterio de evaluación para la fase de Construcción involucra las respuestas a esta cuetiones:

- ¿Es el release del producto estable y suficiente maduro para ser desplegado en la

comunidad de usuarios?

- ¿Están todos los interesados listos para la transición dentro de la comunidad de usuarios?
- ¿Son los actuales gastos de recursos VS. lo planificado todavía aceptables?

La transición tiene que ser pospuesta en un release si el proyecto falla alcanzar este hito.

3.3.5 Artefactos (C)

Artefactos esenciales (en orden de importancia)	Estado al milestone
"El Sistema"	El sistema ejecutable mismo, listo para comenzar una prueba "beta".
RUP: Plan de Desarrollo	Versión inicial desarrollada, revisada y con base-de-línea. • En proyectos más pequeños, esto puede esta encastrado/incrustado en el Plan de Desarrollo de Software.
Modelo de Implementación (y todos los artefactos consitutuyentes, incluyendo los Elementos de Implementación (Componentes)).	Expandido desde que creado durante la fase de Elaboración; todos los elementos de implementación creados para el final de la fase de Construcción.
RUP: Suite de Prueba ("prueba de humo")	Pruebas implementadas y ejecutadas para validar la estabilidad del build para cada uno de los releases de ejecutable creados durante la fase de Construcción.
RUP: Material de Soporte para el "Usuario-Final".	Manuales de Usuario y otros materiales de entrenamiento. • Un borrador preliminar, basado sobre los casos de uso, puede ser necesitado si el sistema tiene un fuerte aspecto de IU.
Plan de Iteración	Plan de iteración para la fase de Transición completada y revisada.
Modelo de Diseño (y todos los artefactos consituyentes)	Actualizado con nuevos elementos de diseño identificados durante la compleción de todos los requerimientos.
RUP: Proceso de Desarrollo	El proceso de desarrollo, incluyendo el caso de desarrollo y cualquier líneas-de-guía y

	templates, ha sido refinado basado sobre la experiencia del proyecto, y está suficientemente definida para la siguiente fase prosiga.
RUP: Infraestructura de Desarrollo	El entorno de desarrollo para la transidción esta en su lugar, incluyendo todas las herramientas y soporte de automatización para el proceso.
RUP: Modelo de Datos	Actualizado con todos los elementos necesitados para soportar la implementación de la persistencia (e.g. tablas, índices, mapeamientos objeto-a-relacional, etc.)
Artefactos opcionales:	
Especificaciones Suplementarias	Actualizado con nuevos requerimientos (si los hubiera) descubiertos durante la fase de Construcción.
Model de Caso-de-Uso (Actores, Casos-de-Uso)	Actualizado con nuevos casos de uso (si los hubiera) descubiertos durante la fase de Construcción.

3.4 Fase: Transición

3.4.1 Meta

El foco de la Fase de Transición es asegurar que el software está disponible para sus usuarios finales.

"La fase de Transición puede abarcar varias iteraciones, e incluye:

→ probar el producto en preparación para el release

→ (y) hacer ajustes menores basado en la realimentación del usuario.

- En este punto en el ciclo-de-vida, la realimentación debe de enfocarse principalmente sobre:
  - → afinamiento del producto
  - → configuración
  - → instalación
  - → (y) temas de usabilidad

  todos las mayores temas estructurales deben de haber sido resueltos mucho mas antes en el ciclo-de-vida del proyecto".

3.4.2 Objetivos:

"Por el final de la Fase de Transición, los objetivos del ciclo-de-vida deben de haber sido satisfechos y el proyecto debe de estar en una posición para ser cerrado.

- En algunos casos, el final del ciclo de vida actual puede coincidir con el arranque de otro ciclo-de-vida del mismo producto, guiando la siguiente generación o versión del producto.
- Para otros proyectos, el final de la Transición puede coincidir con una completa entrega de los artefactos a una tercera parte quien puede ser responsable por:
  - → las operaciones
  - → el mantenimiento
  - → (y) mejoras

  del sistema entregado.

Esta Fase de Transición, oscila desde ser muy sencillo/directo a extremadamente complejo, dependiendo de la clase del producto.

- Un nuevo release de un producto desktop existente puede ser muy simple, mientras que el reemplazo de un Sistema de Control de Tráfico-aereo de una nación puede ser extremadamente complejo.

Actividades realizadas/llevadas a cabo durante una interación en la Fase de Transición dependen de la meta.

Ejemplos:

- cuando arreglando/fijando bugs, implementación y pruebas son usualmente suficientes.
- Si, sin embargo, nuevas características tienen que ser agregadas, la iteración es similar a uno en la fase de Construcción requiriendo análisis & diseño, etc.

La Fase de Transición es entrada cuando una línea-de-base es lo suficiente madura para ser desplegada en el dominio del usuario-final.

- Esto típicamente requiere que algún subconjunto usable del sistema ha sido completado con aceptabale nivel de calidad y documentación usuaria de modo que la transición al usuario provee resultados positivos para todas las partes.

Los objetivos primarios de la Fase de Transición incluyen:

- **Pruebas beta para validar el nuevo sistema** contra las expectativas del usuario.
- **Pruebas beta y operación paralela relativos a un sistema legacy** que ello está reemplazando.
- **Convertir BD's operacionales**
- **Entrenamiento de usuarios y personal de mantenimiento**
- **Extender las fuerzas de marketing, distribución y ventas.**
- **Ingeniería de desplegamiento específica** tal como:
  - → cutover
  - → empaquetamiento y producción comercial
  - → ventas roll-out
  - → entrenamiento de personal de campo

- **Sintonizar activades tal como:**
  - → **arreglar/fijar bugs**
  - → **mejoramiento** para performance y usabilidad

- Valoración de las líneas-de-base de desplegamiento contra la visión completa y los criterios de aceptación para el producto.
- Conseguir auto-soportabilidad del usuario.
- **Conseguir concurrencia de los interesados** que las líneas-de-base de desplegamiento

están completas.

- **Conseguir concurrencia de los interesados** que las líneas-de-base son consistentes con los criterios de la visión.

3.4.3 Actividades esenciales:

Las actividades esenciales de la fase de Transición incluyen:

- Ejecutar planes de desplegamiento
- Finalizar el material de soporte del usuario-final
- Probar el producto entregable en el site de desarrollo
- Crear un release del producto
- Obtener realimentación del usuario
- Sintonizar finamente el producto basado en la realimentación
- Hacer el producto disponible a los usuarios finales

3.4.4 Hito: Release del Producto

El Hito Release del Producto es donde se decide si los objetivos del proyecto fueron satisfechos, y si se debe de arrancar otro ciclo de desarrollo.

- En algunos casos este hito puede coincidir con el fin de la fase de Inicio para el siguiente ciclo.
- El Hito Release del Producto es el resultado de la revisión y aceptación por parte del cliente de los entregables del proyecto (ver Actividad: Revisar Aceptación del Proyecto).

Activity: Project Acceptance Review

Purpose

For the customer to formally review and accept the project deliverables.

Steps

Schedule Project Acceptance Review Meeting

Distribute Meeting Materials

Conduct Project Acceptance Review Meeting

Record Decision

Criterio de Evaluación:

El criterio de evaluación primario para la fase de Transición involucra las respuestas a estas cuestiones:

- ¿Está el usuario satisfecho?

- ¿Son los gastos de recursos actuales VS. los gastos planificados aceptables?

En el Hito Release del Producto, el producto está en producción y el ciclo de mantenimiento post-release comienza.

- Esto puede involucrar arrancar un nuevo ciclo, o algún release de mantenimiento adicional.

3.4.5 Artefactos (T)

Artefactos esenciales (en orden de importancia)	**Estado al milestone**
El Build del Producto	Completa de acuerdo con los requerimientos del producto. • El producto final debe de ser usable por el cliente.
RUP: Material de Soporte para el Usuario-Final	Materiales que asisten al usuario-final en: → aprender → usar → operar → (y) mantener el producto debe de estar completo de acuerdo con los requerimientos.
RUP: Elementos de Implementación	La implementación es completa y con línea-de-base, y los elementos desplegables han sido incorporados en el producto final.
Artefactos opcionales	**Estado en el Hito**
RUP: Suite de Pruebas ("Smoke test")	El suite de prueba desarrollado para validar la estabilidad de cada build puede ser proporcionado/proveído en la situación donde el cliente desea ejecutar un nivel básico de una prueba on-site.
RUP: Empaquetamiento del Producto 'Shrinkwrap'	En el caso de crear un producto shrinkwrap, el contratista necesitará los necesarios artefactos de empaquetamiento para ayudar a vender al por menor el producto.

3.4.6 Resultado:

La iteración final en la fase de Transición culmina en la entrega al cliente de un sistema completo (y artefactos de soporte auxiliares) con la funcionalidad y performance como especificado y demostrador en el test de aceptación.

- El cliente toma propiedad del software después de un test de aceptación exitoso.

Disciplina	**←------- esencia de ------→**	**Mejor Práctica**
...		
Requerimientos	Desarrollar: → una clara visión → (y) un conjunto entendible de requerimientos. • Esto involucra: → analizar el problema → entender las necesidades de los interesados → definir el sistema → (y) gestionar/managing los requerimientos a medida que ellos cambian.	Administrar Requerimientos
Análisis & Diseño	→ definir una arquitectura candidata → refinar la arquitectura → analizar el comportamiento → (y) diseñar componentes del sistema.	Usar Arquitecturas de Componentes
Implementación	→ Incrementalmente construir los componentes del sistema.	Desarrollar Iterativamente
Pruebas	→ probar los componentes del sistema	Desarrollar Iterativamente
...		
UPEDU/RUP: Configuración y Administración del Cambio	→ administrar y controlar el alcance del proyecto, a medida que los cambios ocurren a lo largo del ciclo-de-vida del proyecto, mientras manteniendo la meta de considerar todas las necesidades de los interesados y meeting those, to whatever extent possible.	Controlar Cambios

...		
RUP: Entorno	→ adaptar un proceso para el proyecto	